Int J Adv Manuf Technol (2012) 61:135–147
DOI 10.1007/s00170-011-3699-2

ORIGINAL ARTICLE

Strength modeling and optimizing ultrasonic welded parts of ABS-PMMA using artificial intelligence methods

A. Norouzi · M. Hamedi · V. R. Adineh

Received: 14 February 2011 / Accepted: 10 October 2011 / Published online: 17 November 2011

Abstract The present work deals with modeling and optimization of ultrasonic welding (USW) process parameters including welding time, pressure, and vibration amplitude influencing strength of the welded parts of acrylonitrile butadiene styrene (ABS) and poly(methyl methacrylate) (PMMA) using artificial intelligence (AI) methods. Experiments performed on samples by spot welding workpieces of ABS and PMMA. The experimental data are used for training of artificial neural networks (ANN), adaptive neuro-fuzzy inference systems, and hybrid systems. It is found that ANN had better predictions compared with the other AI methods. The best model was a feed-forward back-propagation network, with uniform transfer functions (TANSIG–TANSIG–TANSIG) and 4/2 neurons in the first/second hidden layers. The best predictor is then presented to genetic algorithm (GA) and particle swarm optimization (PSO), as the fitness function and for optimizing the USW machine parameters. After the optimization, results of this part revealed that GA and PSO have comparable results and the calculated strength increased by 10%, as compared with a non-optimized case. In order to confirm the computational results, validating experiments are performed which their outputs demonstrates good agreement with the optimization result.

A. Norouzi · M. Hamedi
School of Mechanical Engineering, University of Tehran,
Tehran, Iran

V. R. Adineh (✉)
Department of Mechanical Engineering,
Saveh Branch, Islamic Azad University,
Saveh, Iran
e-mail: v.r.adineh@iau-saveh.ac.ir

Keywords Polymer welding · Ultrasonic welding · Artificial neural networks · Adaptive neuro-fuzzy inference systems · Hybrid systems · Genetic algorithm · Particle swarm optimization

1 Introduction

Being used in the automobile industry, welding two dissimilar polymers such as acrylonitrile butadiene styrene (ABS) and poly(methyl methacrylate) (PMMA) is an important process. For instance, front and tail lights of most cars are made up of joining ABS to PMMA. On the other hand, stress cracking is the most frequent cause of failure in the tail lights made from hot tool welding of ABS to PMMA [1]. Internal tensile stress induced by hot tool welding in the range of 230–420 C°, is the source of subsequent cracks. Thus, achieving high weld strength for these types of polymers is significant, especially for the automobile industry.

Ultrasonic plastic welding is one of the main techniques, in which, by conducting ultrasonic vibrations to the upper side of the joint, friction between the part coupled with the sonotrode and the part on the anvil side, causes a local melt and the subsequent pressure joins the polymer chains together. Compared with the conventional welding techniques, ultrasonic welding provides a low-energy bonding technique. Each two thin film polymers having compatibility in joining together can be welded by ultrasonic vibrations, especially amorphous polymers which are preferred to crystalline polymers for joint quality. Furthermore, by the use of high-power ultrasonic energy source, oscillating shear forces break polymer chains and static force produces the high strength solid-state bonds. Besides, the ultrasonic welding technology is an innovative method to produce hybrid joints for multi-material components.

Figure 1 shows a schematic of an ultrasonic plastic welding machine [2]. The most important parameters of the USW process are:

1. The vibration frequency: the frequency which the transducer vibrates with. In most of the machines it is unchangeable
2. The vibrational amplitude of the welding tip
3. The duration of the welding operation (the welding time)
4. The clamping pressure in the weld area

Modeling of systems is of fundamental importance in almost all fields. This is because models enable us to understand a system better, simulate and predict the system behavior and hence help us in optimizing the system parameters. Widely developed linear models are applied in the different areas of engineering, but most of the real time systems are ill defined and uncertain in nature. As a result, system modeling based on the traditional linear systems is not appropriate for complicated systems. Non-traditional systems, namely soft computing which is a collection of methodologies like fuzzy systems, artificial neural networks (ANN), genetic algorithm (GA), particle swarm optimization (PSO), and so forth, were designed to tackle imprecision and uncertainty involved in complex nonlinear systems. The evolution of soft computing techniques has helped in understanding the various aspects of nonlinear systems and thereby making it possible to model them as well as optimization of these systems [3, 4]. Recently, these techniques have found numerous applications for modeling and optimization of various manufacturing and machining processes [5–13].

Since the mid-1960s which the first ultrasonic welding was done, various novel papers have been written investigating the effects of the ultrasonic welding (USW) process parameters on the joint strength and quality. It has been defined that by increasing the weld time, pressure, and amplitude up to a certain value, weld strength will increase. Further increases of the welding parameters will reduce the joint quality and strength [1]. Kirkland [14] has extracted the fundamental formulas needed in frequency selection in ultrasonic plastic welding. Additionally, Tsujino et al. [15] have investigated frequency characteristics of welding machines by using a combination of 90 kHz and 27 or 20 kHz vibration systems and have shown its increasing strength of polyvinyl chloride specimen joints compared with that of 27 kHz plastic welding systems. Moreover, Espinoza et al. [16] have optimized the ultrasonic welding machine parameters such as weld time, hold time, and pressure on the burst strength of the joints of polyurethane

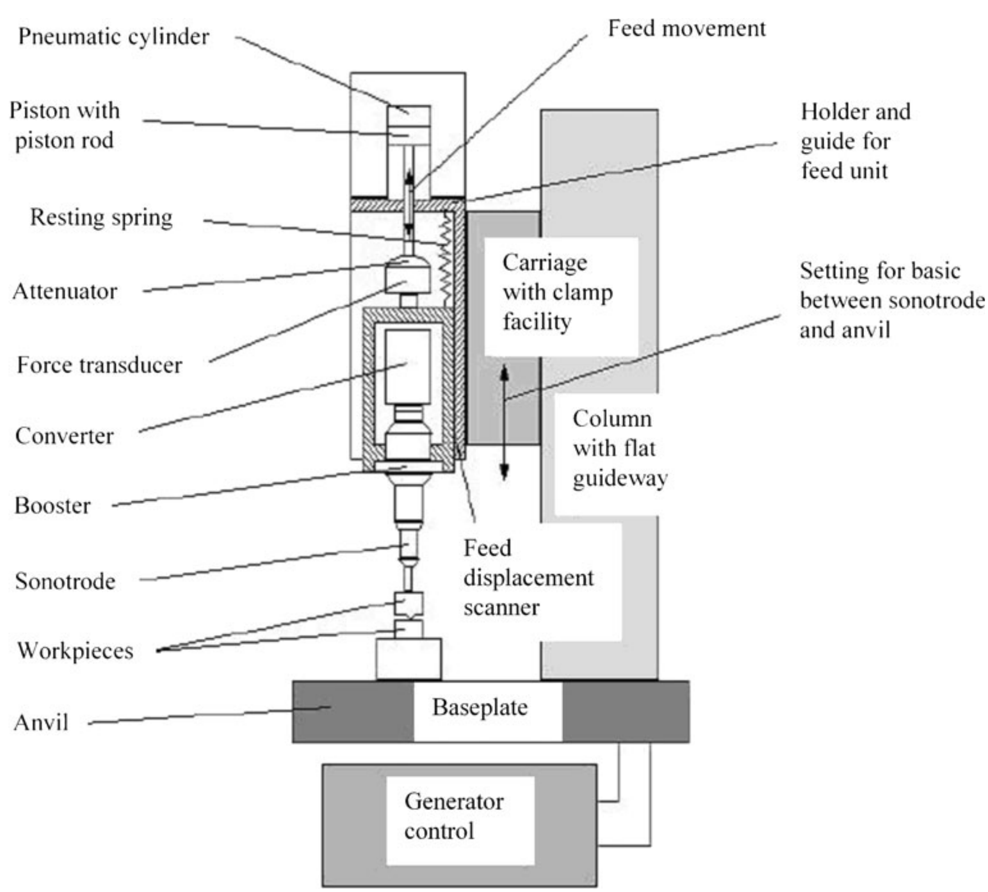

Fig. 1 Schematic of ultrasonic plastic welding machine [2]

specimens using Taguchi robust design. Furthermore, Michaeli et al. [17] have studied optimizing the ultrasonic welding machine parameters of micro parts of polycarbonate by analysis of variance. Compared with the research efforts on the welding of similar materials, only limited research has been done on the ultrasonic welding of dissimilar materials, including ABS and PMMA. Besides, to the best of our knowledge, there is no published result for modeling and optimization of USW machine parameters in joining ABS and PMMA together.

In this research, the effective parameters of the USW process on the weld strength of ABS-PMMA were modeled and optimized using the artificial intelligence (AI) approach. Firstly, the relation between input parameters, that is welding time, pressure and amplitude, and output one, namely strength of the weld, was modeled using ANN, adaptive neuro-fuzzy inference systems (ANFIS), and hybrid systems (HS). All of the models were trained with experimental data which obtained based on a full factorial design. The best model amongst these techniques was selected for the optimization process. Secondly, in order to obtain the maximum weld strength, the selected predictor was introduced as the fitness function to the GA and PSO. Finally, the best optimization result was validated by verifying experiments. Figure 2 illustrates the usage of AI methods for modeling and optimization of the USW process.

The paper is organized as follows: the fundamentals of ANN, ANFIS, HS, GA, and PSO are presented in Section 2. Section 3 illustrates the experimental procedure; implementation of ANN, ANFIS and HS for modeling is described in Section 4; Section 5 explains the GA and PSO implementation for optimization; Section 6 states the model validation step, and Section 7 concludes the paper.

2 Artificial intelligence methods

2.1 Artificial neural networks

ANN is a mathematical model which is inspired by the human biological nerve system and is used to solve complex scientific and engineering problems. The advantage of ANN is the ability to be trained based on experimental/real life data to predict solutions. During the training, an ANN model adjusts itself to establish the relation between the input and the output. In spite of this, an ANN model does not require any explicit formula; but instead it is an implicit model by itself where it can be trained to adopt and adjust itself to perform certain tasks.

In this study, two network types were considered: feed-forward back-propagation (FFBP) and cascade-forward back-propagation (CFBP). The first one is consisted of one input layer, one or several hidden layers and one output layer. Usually, back-propagation learning algorithm (BP) is used to train this network. In the case of the BP algorithm, first the output layer weights are updated. For each neuron of the output layers, there is a constant desired value at first. By this value and the learning rules, the weight coefficient is updated. For some problems, the BP algorithm presents suitable results, while it leads to improper results for others. In some cases, the learning process is upset as a result of getting trapped in local minimum. This is because of the answer, lying at the smooth part of the threshold function. The CFBP network like the FFBP network uses the BP algorithm for updating weights, but the main characteristic of this network is that each layer's neurons are related to all previous layer neurons. In addition, the Levenberg–Marquardt (LM) learning algorithm [18] was utilized for the tuning of the network weights. In addition, the logistic sigmoid (LOGSIG) and tangential sigmoid (TANSIG) activation functions in two hidden layers were used.

2.2 Adaptive neuro-fuzzy inference systems

ANFIS is used for the modeling of nonlinear or fuzzy input and output data, and for the prediction of output according to the input. It applies a combination of the least squares method and the back-propagation gradient descent method for training fuzzy inference system membership function parameters to emulate a given training data set. Functionally, it is equivalent to the combination of neural network and fuzzy inference systems. A comprehensive description regarding ANFIS architecture as well as the learning algorithm of ANFIS can be found in Jang's [19] work.

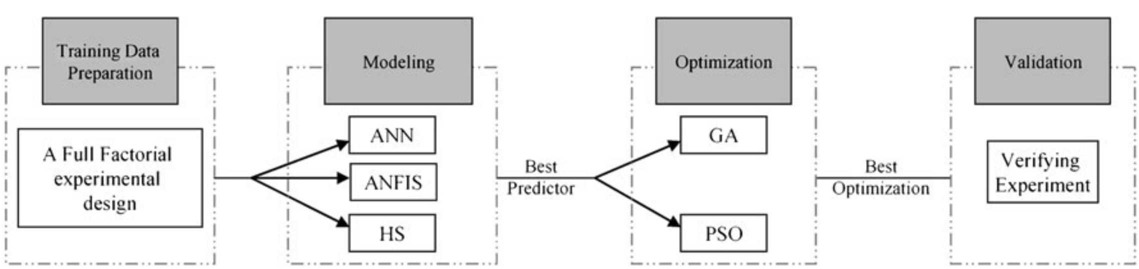

Fig. 2 Application of AI for modeling and optimization of the USW process

2.3 Hybrid systems

Hybrid systems are the methods to solve problems where more than one AI method is employed in [20]. Hybrid systems have been classified as sequential hybrids, auxiliary hybrids and embedded hybrids. In this study, an auxiliary hybrid system; namely, an ANN-PSO hybrids is utilized. In this system which has been used by several authors [21, 22], the gradient descent learning algorithm is replaced by the PSO method to update the weights and biases of ANN. In other words, an ANN-PSO HS makes use of PSO to determine the weights and biases of a multilayer network with the BP learning algorithm.

2.4 Genetic algorithm

GA in particular became popular through the work of John Holland in the early 1970s. A GA is an iterative procedure which borrows the ideas of natural selection and survival of the fittest from natural evolution. GA is categorized as global search heuristics where a population of abstract representations (chromosomes) of candidate solutions (individuals) is simulated as an optimization problem that evolves towards better solutions. Traditionally, solutions are represented in binary as strings of 0 and 1 s, but integers and floating point numbers can also be used. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of each individual in the population is evaluated. Based upon their fitness, multiple individuals are stochastically selected from the current population and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached [23, 24].

2.5 Particle swarm optimization

PSO was originally inspired by social behavior of organisms such as bird flocking and fish schooling. PSO shares many similarities with evolutionary computation techniques such as GA. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. It searches the solution space by assigning velocities v and a neighborhood relationship to the individuals x. An individual in each generation is attracted to the best location in its history (p^h) and to the best location found by its neighborhood (p^n). The classical formulation is given in the following equations:

$$v_i(t+1) = wv_i(t) + \phi_1 r_1 \left(p_i^h - x_i\right) + \phi_2 r_2 \left(p_i^n - x_i\right) \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

The neighborhood type may range between small (local), randomized, and large (global), differing especially in the rate of information distribution. The parameters $\varphi_{1/2}$ control the impact of the attractors, while $r_{1/2}$ are uniform random samples in [0, 1] used as stochastic components. The inertness factor ω controlling the influence of former velocities is usually set<1 for convergence [25, 26].

3 Experimental procedure

In this study, for preparation of training samples, the joining method of two parts of ABS and PMMA together were observed at different working conditions. First of all, the parts were cut from sheets with thickness of 5 mm. In the ASTM G1.1, it is noted that PMMA and ABS have compatibility in joining together [27], but the rigidity of PMMA is higher than ABS. Since the penetration of the welded nuggets into each other is unknown, using an energy director is not possible. In the ultrasonic welding of metals, nuggets are formed almost exactly under the sonotrode [15]. But in the ultrasonic welding of PMMA, due to the rigidity of the polymer and the direction of the tool vibration which differs from the USW of metals, in order to have the nuggets approximately under the sonotrode, workpieces were cut as shown in Fig. 3. Therefore, it could be said that nuggets are exactly under the sonotrode face and the piece's formation does not let the other areas of the piece to be joined by the produced heat of the friction. Experiments were done using 15 kHz ultrasonic welding involving a 10-mm diameter welding tip, as shown in Fig. 4.

At first stage, in order to understand the effects of the welding machine parameters, five levels for welding time, three for pressure and three for amplitude were assigned. Table 1 shows the primal assigned levels for each parameter. After completing the experiments, it was observed that the joints undergoing the first level of welding time had not joint completely and the ones

Fig. 3 Welded parts of ABS and PMMA

Fig. 4 15 KHz ultrasonic welding machine (Maxwide Inc.)

including the fourth and fifth level of the welding time had the least joint quality due to the increase of welding duration. Therefore, these levels were omitted and experiments were done using welding times of 0.4 (level 1) and 0.64 s (level 2).

In the second stage, a full factorial experiment with four repetitions was designed to conduct the experiments. Table 2 shows the designed experiments with relative average strength of four repetitions as well as standard deviation between the four repetitions of same set of process parameters.

4 Strength modeling

In this research, AI methods, which consist of using ANN, ANFIS, and ANN-PSO HS, are used to model the effect of USW machining parameters including welding time, pressure, and vibration amplitude on the strength of welded parts of ABS and PMMA together.

Table 1 Primal levels for experiments

Welding time (s)	Pressure (bar)	Amplitude (μm)
0.24 (level 1) 0.4 (level 2)	2.5 (level 1)	50 (level 1)
0.64 (level 3) 0.88 (level 4)	3.5 (level 2)	55 (level 2)
1.1 (level 5)	4.5 (level 3)	58 (level 3)

Table 2 The average results of joint strength for full factorial design with four repetitions

Run	Welding time (s)	Pressure (bar)	Amplitude (μm)	Average strength (KN)	SD
1	1	1	1	44.50	8.49
2	1	1	2	301.50	13.7
3	1	1	3	528.00	34.6
4	1	2	1	248.50	24.34
5	1	2	2	282.50	12.43
6	1	2	3	210.00	12.63
7	1	3	1	273.00	5.3
8	1	3	2	102.50	18.65
9	1	3	3	232.00	35.3
10	2	1	1	155.00	13.01
11	2	1	2	338.50	40.11
12	2	1	3	358.00	21.97
13	2	2	1	458.00	17.2
14	2	2	2	525.00	17.1
15	2	2	3	679.50	46.05
16	2	3	1	212.50	33.83
17	2	3	2	472.50	17.1
18	2	3	3	630.50	27.4

4.1 Strength modeling using ANN

4.1.1 Pre-processing

To avoid introducing unnecessary bias resulting from variables being measured on the different scales and in order to increase the training velocity, input and output data were normalized. Input data of the first parameter, welding time, were remained un-normalized, while the data of pressure and amplitude were normalized to the boundary of [0,1] using the below min–max equation [28]:

Min–max scaling:

$$X_n = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (3)$$

in which X_n is the normalized parameter, X_i is the real parameter, $X_{\min}$ is the minimum parameter value (of each X_n) and $X_{\max}$ is the maximum parameter value (of each X_n).

The output data were normalized to the boundary of (0.0445, 0.679) using decimal scaling [28]:

Decimal scaling

$$Y' = \frac{Y}{10^n} \quad (4)$$

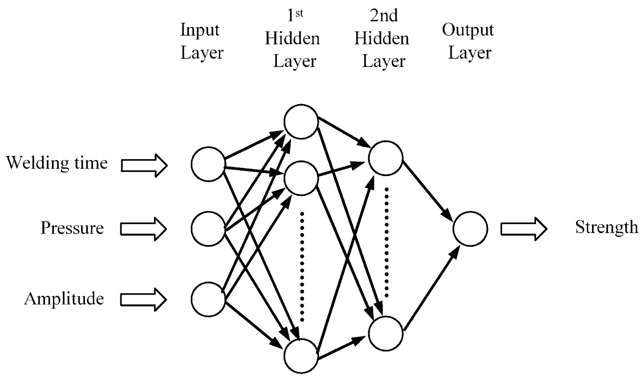

Fig. 5 Topology of considered neural network

where Y' is the normalized target data and Y is the real data. In order to proportionate the normalization boundary of parameters and target data, $n=3$ was determined.

About 70% of the data were used for training, and the remaining was utilized for validation and testing.

4.1.2 Processing

The neural networks toolbox of MATLAB™ was used in this part of the research. Considering and applying three inputs in all experiments, the strength value derived from different conditions, network with three neurons in input layer (welding time, pressure and amplitude) and one neuron in output layer (strength) and two hidden layers were designed. Figure 5 shows the considered neural network topology, Input and output parameters.

One of the great challenges in the modeling of nonlinear systems using ANN is selecting the appropriate number of hidden layer/layers, number of nodes for hidden layers and their transfer functions.

For a simple problem with linear and quadratic approximation, only one hidden layer is needed [29]. By Kolmogorov's theorem, two hidden layers in FFBP are enough for all of the problems, although this theorem was rejected [30]. There are some other networks that have three [31, 32] or more (seven) hidden layers [33] for complicated problems. Additionally, some formulas were given to estimate the number of hidden layers [34]. However, in this study an ANN with two hidden layers was considered as it is adequate in most of the conventional problems [29].

In order to choose an adequate number of nodes for these hidden layers, there is no specific rule; however, there are some guidelines and empirical formulas in various papers and textbooks [29, 35–37]. In this study, a hybrid method for choosing the optimal number of nodes for hidden layers is proposed. This is a hybrid of the empirical formula given by [38] with the increasing method [39]. In this hybrid method, at first the maximum number of nodes for hidden layers is chosen according to the following equation: $h \leq 2_i+1$, where h and i are the number of hidden and input units respectively [38]. Then, according to the increasing method, a network with the minimum number of hidden units was built and then new neurons were added to the network gradually. The hidden units are added until it reaches the maximum possible number of h in the above empirical formula.

According to the above formula, the maximum number of recommended hidden units is seven ones. Consequently, we started from the minimum number of hidden units and continued until reaching the total seven numbers for them. For each combination, the performances of networks were calculated using statistical yardsticks.

For choosing the best transfer functions for hidden units, following strategies were chosen [39]:

Different neurons for several networks at the uniform transfer function for each layer.

Different neurons for several networks at various transfer functions for each layer.

4.1.3 Post-processing

In this paper, the determination coefficient R^2 and root-mean-square error (RMSE) were used according to the

Table 3 Results for different neurons and hidden layers of a FFBP for several networks at the uniform transfer function for layers

Net	Topology	Transfer function	R^2			RMSE		
			Train	Val	Test	Train	Val	Test
1	2-2-1	T-T-T	0.8988	0.9981	0.8889	0.0692	0.0643	0.0591
2	3-2-1	L-L-L	0.9472	1	0.9971	0.0394	0.0775	0.0717
3	4-2-1	T-T-T	0.9785	0.9994	0.9989	0.0271	0.0294	0.0268
4	5-2-1	L-L-L	0.9657	0.9855	0.9976	0.0327	0.0382	0.0693
5	3-3-1	T-T-T	0.9181	0.945	0.9996	0.0527	0.0434	0.0529
6	4-3-1	L-L-L	0.971	0.9126	0.9534	0.0292	0.0658	0.0602

Table 4 Results for different neurons and hidden layers of a FFBP for several networks at the various transfer function for layers

Net	Topology	Transfer function	R^2			RMSE		
			Train	Val	Test	Train	Val	Test
1	2-2-1	T-L-T	0.8795	0.857	0.9288	0.0609	0.0839	0.0456
2	3-2-1	L-T-L	0.9498	0.9683	0.9933	0.0384	0.0774	0.0426
3	4-2-1	T-T-L	0.9114	0.8841	0.9957	0.0616	0.0552	0.0714
4	5-2-1	L-L-T	0.8905	0.9585	1	0.0659	0.0727	0.0784
5	3-3-1	L-T-T	0.8868	0.9837	0.9712	0.0593	0.0493	0.051
6	4-3-1	T-L-L	0.9486	0.8413	0.9812	0.0401	0.0709	0.0789

following equations as a yardstick for performance measurement:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(S_k - T_k)^2} \tag{5}$$

Where T denotes the number of data patterns, S_k is the network output of the kth pattern and T_k is the target output of the kth pattern (experimental output data).

In addition, the determination coefficient is measured as follows:

$$R^2 = 1 - \frac{\sum_{k=1}^{T}[S_k - T_k]}{\sum_{k=1}^{T}[S_k - T_m]}, T_m = \frac{\sum_{k=1}^{T} S_k}{T} \tag{6}$$

When the RMSE is at the minimum and R^2 is high, ($R^2 \geq 0.8$); a model can be evaluated as very good [40]. Furthermore, the fact that the RMSE error values are very close to each other, is an indication of sufficient regularization [41].

4.1.4 Results and discussion of ANN results

Both strategies were used for FFBP and CFBP networks with the LM learning algorithm. The best results for these networks and algorithms are shown in Table 3, 4, 5, and 6.

Tables 3, 4, 5, and 6 show that increasing the number of hidden units leads to poor generalization. In Table 3, it is concluded that the LOGSIG activation function had better outcomes than the TANSIG. However, the best result in this strategy belongs to the network with a topology of 4-2-1 with the TANSIG transfer function which produces 0.9785, 0.9994, and 0.9976 for the determination coefficient of the train, validation, and test sets, respectively, with an RMSE of 0.0271, 0.0294 and 0.0268. Table 4 shows that the network with a topology of 3-2-1 and LOGSIG–TANSIG–LOGSIG transfer functions had better results compared with the others. Furthermore, it can be concluded that for the FFBP networks, the strategy of uniform transfer functions for all layers had better performance than the second one. For CFBP networks with uniform activation functions, according to Table 5, it is found that the second network had the best predictions. Additionally, it is seen that with the LOGSIG transfer function, the CFBP networks have poor generalization compared with the TANSIG ones. Besides, Table 6 demonstrates that the second network with a topology of 3-2-1 by LOGSIG–TANSIG–LOGSIG transfer functions had the best predictions. For the CFBP network, generally speaking, the second strategy had a better performance.

Finally, amongst all of the four best networks, the FFBP network with a 4-2-1 topology and TANSIG activation functions had the best predictions; thus, it is selected from this subsection as the best evaluator.

Table 5 Results for different neurons and hidden layers of a CFBP for several networks at the uniform transfer function for layers

Net	Topology	Transfer function	R^2			RMSE		
			Train	Val	Test	Train	Val	Test
1	2-2-1	T-T-T	0.9777	0.9956	0.957	0.026	0.0361	0.0389
2	3-2-1	L-L-L	0.9918	0.9853	0.9974	0.0173	0.046	0.0462
3	4-2-1	T-T-T	0.9817	0.976	0.9997	0.0269	0.0691	0.0677
4	5-2-1	L-L-L	1	0.9957	0.9789	0.0008	0.0455	0.0774
5	3-3-1	T-T-T	0.9982	0.9508	0.9987	0.008	0.038	0.0571
6	4-3-1	L-L-L	1	0.9829	0.9911	0	0.0506	0.0749

Table 6 Results for different neurons and hidden layers of a CFBP for several networks at various transfer function for layers

Net	Topology	Transfer function	R^2			RMSE		
			Train	Val	Test	Train	Val	Test
1	2-2-1	T-L-L	0.9796	0.9984	0.9672	0.0279	0.0314	0.0249
2	3-2-1	L-T-L	0.9855	0.9811	0.9808	0.0207	0.0268	0.0186
3	4-2-1	T-T-L	0.9547	0.9563	0.9644	0.038	0.0388	0.0337
4	5-2-1	L-L-T	0.9892	0.9577	0.9652	0.0181	0.0563	0.0676
5	3-3-1	L-T-T	0.9538	0.9792	0.9851	0.037	0.0499	0.0207
6	4-3-1	T-L-T	1	0.9746	0.9779	0	0.0259	0.0431

4.2 Strength modeling using ANFIS

4.2.1 Pre-processing

In this subsection, the normalizing of data is similar to Section 4.1.1., but there is further processing for dividing the data into the train, validation, and test sets. This is because ANFIS is very sensitive to the input variable selection for learning; that is, one of the great challenges in the modeling of nonlinear systems using ANFIS is selecting the important input variables from all possible input variables. Consequently, it is necessary to do input selection that finds the priority of each candidate inputs and uses them accordingly [42, 43]. For this purpose, the Jang's method was selected [43]. The utilized input selection method is based on the assumption that "the ANFIS model with the smallest RMSE after one epoch of training has a greater potential of achieving a lower RMSE when given more epochs of training" [43]. Using a written code in MATLAB™, a set of train/validation/test sets were made randomly, and then were trained for one epoch. The smallest one was used for further processing. These optimum sets are labeled as the best sets in Table 7, and the corresponding number is related to the values of RMSE after one epoch of training, for the best set amongst 1,000 runs.

4.2.2 Processing

The Fuzzy Logic Toolbox of MATLAB™ was used for this part of the research. In order to find the optimum ANFIS parameters for training, two different strategies were utilized:

- Various kinds of membership functions (MFs) with equal number of MFs
- Various kinds of MFs with different number of MFs.

Since it has been proven that the hybrid algorithm is highly efficient in the training of ANFIS [19], it was selected as the learning algorithm. Besides, Gaussian (gaussmf) and bell shape (gbell) MFs were selected as MF types.

4.2.3 Post-processing

The same criteria that was used for ANNs, is applied for ANFIS.

Table 7 Various kinds of membership functions with equal/different number of membership functions

No.	No. of MFs	Best set	MFs type	R^2			RMSE		
				Train	Val	Test	Train	Val	Test
1	2-2-2	0.0178	gbelmf	0.9913	0.7256	0.588	0.0169	0.1945	0.2874
		0.0154	gaussmf	0.9948	0.2065	0.1379	0.0149	0.1757	0.3269
2	3-3-3	0	gbelmf	1	0.1834	0.8941	0	0.5223	0.4897
		0	gaussmf	1	0.8738	0	0	0.5157	0.5103
3	3-2-2	0.0185	gbelmf	0.9925	0.9929	0.0759	0.0152	0.1461	0.2932
		0.0241	gaussmf	0.9429	0.7141	0.9231	0.0398	0.0465	0.1642
4	2-3-2	0.007	gbelmf	0.9968	0.9643	0.0143	0.007	0.3714	0.2985
		0.006	gaussmf	1	0.9799	0.791	0.0001	0.3226	0.1196
5	2-2-3	0	gbelmf	1	0.8241	0.48	0	0.1854	0.1373
		0	gaussmf	1	0.1684	0.9468	0	0.2432	0.2727

Table 8 Results for different neurons and hidden layers of a FFBP for several networks at the uniform transfer function for layers

Net	Topology	Transfer function	R^2		RMSE	
			Train	Test	Train	Test
1	2-2-1	T-T-T	0.8697	0.7056	0.0618	0.1534
2	3-2-1	L-L-L	0.979	0.8241	0.025	0.1336
3	4-2-1	T-T-T	0.909	0.6177	0.0517	0.1131
4	5-2-1	L-L-L	0.9915	0.637	0.0163	0.1671
5	3-3-1	T-T-T	0.8678	0.8995	0.0625	0.212
6	4-3-1	L-L-L	0.9652	0.6826	0.0335	0.1289

4.2.4 Results and discussion of ANFIS results

First it should be noted from Table 7 that increasing the number of membership functions rapidly leads to the overtraining of the ANFIS. Next, it is seen that the second strategy, i.e. different number of MFs, had better consequences compared with the first one. Finally, the best ANFIS predictor is an ANFIS with an order of 3-2-2 number of membership functions for the first, second and third inputs respectively, with Gaussian MFs. Although there are some published results that ANFIS had better predictions compared with the ANN [44], but it is seen that in this study, ANN provides better results.

4.3 Strength modeling using the ANN-PSO hybrid system

4.3.1 Pre-processing

The same normalized data for the ANN section were used for this part, but validation data were added to the test data for measuring the performance of HS.

4.3.2 Processing

For this subsection, the ANN-PSO code given by [45] was utilized. Like Section 4.1.2., both strategies were used for FFBP and CFBP networks, but here PSO is used instead of the LM learning algorithm for updating the weights and biases.

4.3.3 Post-processing

The same criteria used for ANNs and ANFIS, were applied for this part.

4.3.4 Results and discussion of ANN-PSO HS results

According to the obtained results, it is evident that in this study, PSO training of ANN had lower performance than the LM algorithm. Table 8 shows that a 3-2-1 topology with LOGSIG transfer function had better results. For the FFBP network with various transfer functions, a 4-3-1 topology not only does not result in over fitting, but also produced suite predictions compared with other networks of Table 9. For the CFBP network with the same activation functions (Table 10), the network with three hidden units for each hidden layer with the TANSIG function had the best predictions. For networks of Table 11, a 3-3-1 topology had the best results. By Tables 8, 9, 10, and 11, firstly it is concluded that the CFBP network with a 3-3-1 topology and various types of activation functions when the PSO algorithm is used for training, had the best predictions. Secondly, contrary to the training with LM, increasing the hidden units resulted in better predictions.

4.4 Results and discussion of Strength Modeling

AI methods were implemented for modeling strengths of welded parts. The best predictor of the ANN method

Table 9 Results for different neurons and hidden layers of a FFBP for several networks at the various transfer function for layers

Net	Topology	Transfer function	R^2		RMSE	
			Train	Test	Train	Test
1	2-2-1	T-L-T	0.7944	0.7234	0.0777	0.2041
2	3-2-1	L-T-L	0.9876	0.8264	0.0192	0.1544
3	4-2-1	T-T-L	0.9993	0.7889	0.0045	0.3156
4	5-2-1	L-L-T	0.988	0.7239	0.0195	0.2085
5	3-3-1	L-T-T	0.9568	0.669	0.0317	0.4103
6	4-3-1	T-L-L	0.9999	0.8283	0.0015	0.1488

Table 10 Results for different neurons and hidden layers of a CFBP for several networks at the uniform transfer function for layers

Net	Topology	Transfer function	R^2		RMSE	
			Train	Test	Train	Test
1	2-2-1	T-T-T	0.9778	0.903	0.0262	0.2492
2	3-2-1	L-L-L	0.9734	0.7447	0.0279	0.1524
3	4-2-1	T-T-T	0.9999	0.8164	0.0021	0.3305
4	5-2-1	L-L-L	0.9975	0.7367	0.009	0.1805
5	3-3-1	T-T-T	0.9594	0.8687	0.0384	0.2042
6	4-3-1	L-L-L	0.9733	0.8029	0.028	0.1039

is a FFBP network with 4/2 hidden layers for the first and second layers and TANSIG/TANSIG/TANSIG transfer functions. The best ANFIS model is an ANFIS with 3-2-2 number of MFs for the first, second and third inputs, with Gaussian MFs. For ANN-PSO HS, the CFBP network with a 3-3-1 topology and various types of activation functions had the best predictions. In comparison with the resulted values of R^2 and RMSE from these evaluators, it is established that the ANN method had better predictions amongst other methods. This evaluator is selected for the optimization process, namely to introduce as the fitness function to the GA and PSO optimization algorithms.

5 Optimization of USW machine parameters

The goal of optimization is to find values of the variables that minimize or maximize the objective function while satisfying the constraints. For this physical system, the optimization problem can be expressed as the follows:

- Find—welding time, pressure and amplitude
- Maximize—weld strength (a function of welding time, pressure and amplitude)
- Subject to the following variable bounds—$0.4 \le$ welding time ≤ 0.64 and $0 \le$ pressure and amplitude ≤ 1

The bounds of these variables are in the range of valid neural network modeling boundaries. Since the pressure and amplitude parameters were normalized to the boundary [0, 1] (the welding time was not normalized) and the designed ANNs were trained with these normalized parameters, the only parameters that belong to these boundaries could have valid ANN outputs [46]. In other words, these boundaries are constraints of this optimization problem. As noted before, the best founded predictor was presented to the GA and PSO as the fitness function.

5.1 Optimization by use of GA

For this section, the GA toolbox of MATLAB™ was selected for implementing the GA for optimization. In the GA, the deployment of the optimal solution search requires the tuning of some features related with the GA; for example population size, selection method and crossover functions, mutation rate, migration, etc. Although some general guidelines about such selections exist in the relevant literature [23, 47, 48], the optimal setting is strongly related to the design problem under consideration. Thus, in most of the cases it is obtained through the combination of the designer's experience and experimentation. Table 12 shows optimized values of USW process parameters using GA.

Table 11 Results for different neurons and hidden layers of a CFBP for several networks at the various transfer function for layers

Net	Topology	Transfer function	R^2		RMSE	
			Train	Test	Train	Test
1	2-2-1	T-L-T	0.9914	0.6761	0.0158	0.2467
2	3-2-1	L-T-L	0.9944	0.6158	0.013	0.1544
3	4-2-1	T-T-L	0.995	0.7933	0.0383	0.1869
4	5-2-1	L-L-T	0.9049	0.6308	0.0533	0.1667
5	3-3-1	L-T-T	0.965	0.947	0.0326	0.2669
6	4-3-1	T-L-L	0.9998	0.705	0.0027	0.168

Table 12 Optimization results of GA and PSO

Method	Optimum parameters (normalized)			Optimization result (normalized calculated strength)
	Welding time (s)	Pressure (bar)	Amplitude (μm)	
GA	0.616	0.645	1	0.7465
PSO	0.614	0.648	1	0.7465

5.2 Optimization by use of PSO

The PSO toolbox in MATLAB™ released by [49] was utilized for this part of the study. PSO has been applied successfully to a wide variety of search and optimization problems and it has been proven to be a powerful competitor to the other evolutionary algorithms such as GA [26, 50]. Moreover, a simple algorithm, easy to implement and few parameters to adjust are other advantages of PSO algorithms over other ones. Furthermore, there are some strategies and few empirical rules that can be found in the manuscripts [51–54], which could be worked out to guide the effective choice of parameters. Table 12 demonstrates optimization results calculated by PSO.

5.3 Results and discussion of optimization results

In this paper, GA and PSO methods were used to set the parameters of USW process such as welding time, pressure and amplitude, to obtain the maximum weld strength. The best result of each method is shown in Table 12. It is observed that GA and PSO results are comparable to each other. Although there are slight differences in the optimum values of welding time and pressure for GA and PSO calculations, but it leads to differences of the strength value in the order of 10^{-5}, which in this case is insignificant. The maximum value of non-optimized experimental outputs for strength value, depicted in Table 2, is 679.50 kN that increases to 747.65 kN after the optimization. An improvement of 10% in strength is gained through optimization.

6 Validation

In accordance with the number of repetitions in the experimental procedure, four validation experiments with the optimized set of process parameters, as in Table 12, were conducted to verify the calculations. Table 13 lists the non-optimized, optimized and validation experiment results with the average output of validation experiments as 730 kN (737, 755, 708, and 720 kN for each individual output) and 755 kN being the maximum value. Compared with the non-optimized results, an 8% and 11% gain in the strength of the weld is achieved.

It is seen that there is a difference between the results of GA and PSO optimization with the validation experiments. This can be attributed to different sources with the most important ones being the inaccuracy in making the specimens and errors in the experimental measurements affecting the accuracy of the data gathered for training the models. This leads to decreasing the accuracy of the modeling process and therefore undermining the accuracy of the fitness function utilized for GA and PSO.

7 Conclusions

In this article, application of AI methods for modeling and optimization of a manufacturing process, that is USW of ABS and PMMA, was presented. To our knowledge, there is no published result for modeling or optimizing USW of ABS and PMMA. The modeling process started with preparation of the training data according to a full factorial

Table 13 Non-optimized, optimized, and validation experiment results

Method/experiment	Parameters			Objective function Strength (KN)	Improvement vs. non-optimized case (%)
	Welding time (s)	Pressure (bar)	Amplitude (μm)		
Non-optimized result	0.64	3.5	58	679	–
Normalized GA and PSO	0.61	0.64	1	0.747	–
De-normalized GA and PSO	0.61	3.79	58	747	10
Average of four validation experiments	0.61	3.8	58	730	8
The maximum validation experiments value	0.61	3.8	58	755	11

experimental design being used for training the models based on ANN, ANFIS and ANN-PSO HS methods. By choosing a proper strategy for setting the variables of the modeling methods, the modeling results revealed that ANN had better performance than ANFIS as well as HS methods. The best model was a FFBP ANN with 4/2 hidden units in the first/second hidden layers, with TANSIG transfer functions for all layers. Then, the optimization process was initiated by presenting the best evaluator to the optimization algorithms, GA and PSO, as the fitness function. It is found that GA and PSO, in this case, had the same performance since the optimization results of both of them were identical. Both algorithms were able to improve the weld strength value by 10%. Finally, in order to validate the calculations, verification experiments were performed. The results of verifying experiments displayed good agreement with the optimizing computations. Comparing the values of measured weld strengths in the non-optimized-cases with the average output of verifying experiments shows an 8% increase. The optimization methodology developed in this research can be used to model and optimize any general manufacturing process. Also the optimized results for USW process can be used to increase the quality of the relevant processes such as welding the front and tail lights of most of cars that are made up of ABS and PMMA in the automobile industries.

References

1. Troughton MJ (2008) Handbook of plastic joining: a practical guide, 2nd edn. Plastics Design Library, Norwich, NY
2. Devin J (1981) Ultrasonic plastic welding basics. Welding Journal 29–33
3. Kamiya A, Ovaska S, Roy R, Kobayashi S (2005) Fusion of soft computing and hard computing for large-scale plants: a general model. Appl Soft Comput 5(3):265–279
4. Kayak O, Zadeh L (1998) Fuzzy inference systems: a critical review, computational intelligence soft computing and fuzzy-neurointegration with applications. Springer, Berlin
5. Pradhan MK, Biswas CK (2010) Neuro-fuzzy and neural network-based prediction of various responses in electrical discharge machining of AISI D2 steel. Int J Adv Manuf Technol 50:591–610
6. Lee ZJ, Ying KC, Chen SC, Lin SW (2010) Applying PSO-based BPN for predicting the yield rate of DRAM modules produced using defective ICs. Int J Adv Manuf Technol 49:987–999
7. Tavakkoli-Moghaddam R, Azarkish M, Sadeghnejad-Barkousaraie A (2010) Solving a multi-objective job shop scheduling problem with sequence-dependent setup times by a Pareto archive PSO combined with genetic operators and VNS. Int J Adv Manuf Technol. doi:10.1007/s00170-010-2847-4
8. Costa A, Celano G, Fichera S (2010) Optimization of multi-pass turning economies through a hybrid particle swarm optimization technique. Int J Adv Manuf Technol. doi:10.1007/s00170-010-2861-6
9. Ghosal S, Chaki S (2010) Estimation and optimization of depth of penetration in hybrid CO2 LASER-MIG welding using ANN-optimization hybrid model. Int J Adv Manuf Technol 47:1149–1157
10. Sha DY, Hung Lin H (2009) A particle swarm optimization for multi-objective flowshop scheduling. Int J Adv Manuf Technol 45:749–758
11. António CAC, Davim JP, Lapa V (2008) Artificial neural network based on genetic learning for machining of polyetheretherketone composite materials. Int J Adv Manuf Technol 39:1101–1110
12. Sarkar S, Mitra S, Bhattacharyya B (2006) Parametric optimisation of wire electrical discharge machining of titanium aluminide alloy through an artificial neural network model. Int J Adv Manuf Technol 27(5):501–508
13. Lee WM, Liao YS (2007) Adaptive control of the WEDM process using a self-tuning fuzzy logic algorithm with grey prediction. Int J Adv Manuf Technol 34(5):527–537
14. Kirkland TR (2001) The implications of the fundamental formulas for frequency selection in ultrasonic plastics welding. In: 31st annual symposium of Ultrasonic Industry Association Atlanta, Georgia, USA,
15. Tsujino J, Hongoh M, Yoshikuni M, Hashii H, Ueoka T (2004) Welding characteristics of 27, 40 and 67 kHz ultrasonic plastic welding systems using fundamental- and higher-resonance frequencies. Ultrasonics 42(1–9):131–137
16. Alejandro A. Espinoza JER (2004) An optimization study of the ultrasonic welding of thin film polymers. In: Proceedings of DETC 04 ASME 2004.
17. Michaelia W, Haberstrohb E, Hoffmann WM (2008) Ultrasonic welding of micro plastic parts. Multi-material micro manufacture. Cardiff University, Cardiff, UK
18. Hagan MT, Menhaj MB (1994) Training feed forward networks with the marquardt algorithm. IEEE Transactions on Neural Networks 5(6):989–993
19. Jang JSR (1993) ANFIS: Adaptive-Network-based fuzzy inference systems. IEEE Transactions on Systems, Man, And Cybernetics 23(3):665–685
20. Rajasekaran S, Pai GAV (2005) Neural networks, fuzzy logic, and genetic algorithms synthesis and application. Prentice-Hall, New Delhi
21. Li T-S, Hsu C-M (2010) Parameter optimization of sub-35 nm contact-hole fabrication using particle swarm optimization approach. Expert Syst Appl 37(1):878–885
22. Majhi B, Panda G (2011) Robust identification of nonlinear complex systems using low complexity ANN and particle swarm optimization technique. Expert Syst Appl 38(1):321–333
23. Goldberg DE (1989) Genetic algorithm in search optimization and machine learning. Addison-Wesley, Boston, MA
24. Xing WX (1998) Modern optimization algorithm. Tinghua University Press, Beijing
25. Eberhart RC, Shi Y (2001) Particle swarm optimization development, applications and resources. In: Congress on Evolutionary Computation, Piscataway, NJ. IEEE Press, pp 81–86
26. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE International Conference on Neural Networks, Perth, WA, Australia. IEEE Service Center, pp 1942–1948
27. AWS (2006) Guide to ultrasonic assembly of thermoplastics. G1.1M/G1.1. American Welding Society, Miami
28. Myatt GJ (2007) Making sense of data a practical guide to exploratory data analysis and data mining. Wiley, Hoboken, NJ
29. Trenn S (2008) Multilayer perceptrons: approximation order and necessary number of hidden units. IEEE Transactions On Neural Networks 19(5):836–844
30. Nakamura M, Mines R, Kreinovich V (1993) Guaranteed intervals for Kolmogorov's theorem (and their possible relation to neural networks). Interval Computations 3:183–199

31. Sapuan SMMI (2010) Composite materials technology neural network applications. CRC Press, Florida
32. Guo WW (2010) A novel application of neural networks for instant iron-ore grade estimation. Expert Syst Appl 37(12):8729–8735
33. Mohanty JRVB, Parhi DRK, Ray PK (2009) Application of artificial neural network for predicting fatigue crack propagation life of aluminum alloys. Archives of Computational Materials Science and Surface Engineering 1:133–138
34. Atik K, Aktas A, Deniz E (2010) Performance parameters estimation of MAC by using artificial neural network. Expert Syst Appl 37(7):5436–5442
35. Jung J-R, Yum B-J (2011) Artificial neural network based approach for dynamic parameter design. Expert Syst Appl 38 (1):504–510
36. Khosravi A, Nahavandi S, Creighton D (2010) A prediction interval-based approach to determine optimal structures of neural network metamodels. Expert Syst Appl 37(3):2377–2387
37. Ashrafi HR, Jalal M, Garmsiri K (2010) Prediction of load-displacement curve of concrete reinforced by composite fibers (steel and polymeric) using artificial neural network. Expert Syst Appl 37(12):7663–7668
38. Swingler K (1996) Applying neural networks: a practical guide. Academic Press, San Diego, CA
39. Adineh VR, Aghanajafi C, Dehghan GH, Jelvani S (2008) Optimization of the operational parameters in a fast axial flow CW CO2 laser using artificial neural networks and genetic algorithms. Optics & Laser Technology 40(8):1000–1007
40. Kasabov NK (1998) Foundations of neural networks, fuzzy systems and knowledge engineering. A Bradford book, 2nd edn. MIT Press, Cambridge, MA
41. Kermani BG (2007) Modeling oligonucleotide probes for SNP genotyping assays using an adaptive neuro-fuzzy inference system. Sensors and Actuators B 121:462–468
42. Stephen LC (1996) Selecting input variables for fuzzy models. J Intelligent & Fuzzy Systems 4(4):243–256
43. Jang JSR (1996) Input selection for ANFIS learning. In: The IEEE international conference on fuzzy systems, New Orleans, LA, USA. pp 1493–1499
44. Das A, Maiti J, Banerjee RN (2010) Process control strategies for a steel making furnace using ANN with bayesian regularization and ANFIS. Expert Syst Appl 37(2):1075–1085
45. Brige B TPSOt (2003) A particle swarm optimization toolbox for use with matlab. In: Proceedings of the 2003 IEEE swarm intelligence symposium, Indianapolis, Indiana, USA. pp 182–186
46. Deemuth H, Beale M, Hagan M (2007) Neural network toolbox 5 user's guide. The Mathworks Inc., Natick, MA
47. Mendi F, Baskal T, Boran K, Boran FE (2010) Optimization of module, shaft diameter and rolling bearing for spur gear through genetic algorithm. Expert Syst Appl 37(12):8058–8064
48. Goldberg DE (2002) The design of innovation. Kluwer, Norwell
49. Evers GI, Ghalia MB (2009) Regrouping particle swarm optimization: a new global optimization algorithm with improved performance consistency across benchmarks. In: IEEE international conference on systems, man, and cybernetics, San Antonio, TX. pp 3901–3908
50. Yoshida H, Kawata K, Fukuyama Y, Nakanishi Y A (1999) Particle swarm optimization for reactive power and voltage control considering voltage stability. In: Proc. Int. Conf. Intell. Syst. Appl. Power Syst., Rio de Janeiro, Brazil. pp 117–121
51. Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 85:317–325
52. Clerc M (2006) Particle swarm optimization. British Library Cataloguing-in-Publication Data
53. Bergh FVD (2002) An analysis of particle swarm optimizers. Department of Computer Science, University of Pretoria, Pretoria, South Africa
54. Jiang M, Luo YP, Yang SY (2007) Particle swarm optimization—stochastic trajectory analysis and parameter selection. In: Felix TSC (ed) Swarm intelligence focus on ant and particle swarm optimization. I-TECH Education and Publishing, Wien